

C3A - Git

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f4b26a18-09fb-4668-bcbe-b9ea069ab599/git-cheatsheet-ESP-white.pdf>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c59d3617-496b-4b0e-b46b-926da4d67d45/git-cheatsheet-ESP-white.pdf>

¿Qué vamos a ver en esta clase?

En la primera clase de este módulo trabajaremos con una herramienta que nos acompañará a lo largo de la cursada: Git. Es muy útil para compartir proyectos con distintas personas, tener varias versiones de un mismo proyecto y poder guardar proyectos en la nube. Es por esto que es una de las más requeridas por el mercado a la hora de trabajar con control de versiones y de forma colaborativa. Como muchas de las herramientas que veremos a lo largo de la cursada, Git requiere de práctica y tiene muchos accesorios que agregan

funcionalidades, pero en esta clase nos centraremos en sus cuestiones principales.

Compartimos la lista de cuestiones que abordaremos en esta clase:

- Introducción a Git
- Instalación de Git
- Uso de GitHub y creación de repositorios remotos
- Creación de repositorios locales
- Agregado de archivos al repositorio local
- Confirmación de archivos en el repositorio local
- Subida de archivos al repositorio remoto
- Bajada de archivos desde un repositorio remoto
- ¿Cómo resolver conflictos?

Introducción a Git

¿Cómo se trabaja en forma colaborativa en un ambiente profesional? ¿Nos enviamos mails con código? ¿Mensajes de WhatsApp? ¿Usamos Google Drive? Mmm... ¿suena raro, no?

Definitivamente no usamos nada de todo esto, pero utilizamos una herramienta que tiene muchos de los beneficios que nos dan las que mencionamos anteriormente: trabajar en grupo de forma colaborativa, comunicar cambios, manejar distintas versiones. Como dijimos al principio del módulo, vamos a conocer una herramienta que nos va a servir durante toda la cursada.

En el mundo de desarrollo de aplicaciones, web o mobile, siempre nos encontramos con la imperativa necesidad de:

- Backup de nuestros archivos continuamente actualizados.
- La posibilidad de compartir nuestro trabajo con los demás colaboradores de nuestro equipo de trabajo.

¿Que es en sí Git?


Es un software de control de versiones pensado para la eficiencia y confiabilidad del mantenimiento de aplicaciones cuando estas tienen un gran numero de archivos de código fuente. Su propósito es llevar registro de los cambios de los archivos y coordinar el trabajo que varias personas realizan sobre los mismos.


Instalación de Git

¿Cómo saber si Git está instalado? Si estamos usando una Mac, lo más probable es que ya esté instalado. En caso de tener Windows, debemos instalar Git Bash. Los usuarios de Linux necesitamos buscar las instrucciones particulares para nuestra distribución.

Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn

 <https://git-scm.com/>



Creando nuestro primer repositorio local

Un repositorio, sea local o remoto, es un almacén de archivos en donde se irán almacenando tus archivos en pequeños paquetes llamados commits.

- Repositorio local: es el que tiene todos los archivos (que hayas guardado en él) en la computadora.
- Commits: son los paquetes que nos van a permitir ir haciendo un seguimiento de los cambios que vamos realizando, dado que cada uno de ellos tiene una timestamp, o fecha de creación, y un autor. Los commits van a ser nuestro historial de cambios que se fueron haciendo en el proyecto.

```
git init // crea el repositorio
git config user.name "nombre de usuario" // agrega nuestra identidad
git config user.email "nombreUsuario@email.com" // agrega nuestro email
git remote add origin https://github.com/DH/RepoRemoto // apunta al repositorio remoto
```

Agregando archivos al repositorio

Para que Git pueda llevar un control de las modificaciones realizadas en un archivo tenemos que indicarle qué archivos queremos que mire. Pero... ¿Cómo se hace eso?

```
git add nombre_archivo // agrega solo el archivo nombre_archivo
git add . // agrega todos los archivos
git status // seguimiento del estado de los archivos
```

Confirmando archivos

Cuando trabajamos con archivos, estamos acostumbrados a que estos se guarden en forma automática o, a pedirle al programa que lo guarde (el famoso Ctrl + S). Confirmar las modificaciones en Git es de suma importancia ya que nos permite establecer un punto de control. ¡Veamos cómo hacerlo!

- Commits: confirmación de agregado de archivos.

```
git add . // agrega los archivos
git commit -m "mensaje" // comitea los cambios hechos
git log // Historial de los cambios del proyecto
```